



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

Address: COMMISSIONER FOR PATENTS

P.O. Box 1450

Alexandria, Virginia 22313-1450

www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/786,772	02/25/2004	Ross Bunker	10440200234	2871
74739 7590 07/08/2009 Squire, Sanders & Dempsey L.L.P. Oracle International Corporation 8000 Towers Crescent Drive 14th Floor Vienna, VA 22182				
EXAMINER				
PHAM, MICHAEL				
ART UNIT		PAPER NUMBER		
2167				
MAIL DATE		DELIVERY MODE		
07/08/2009		PAPER		

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary

Application No.

10/786,772

Applicant(s)

BUNKER ET AL.

Examiner

MICHAEL PHAM

Art Unit

2167

Period for Reply -- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 30 April 2009.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-4, 6-11, 13-18, 20, 21 and 29-34 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-4, 6-11, 13-18, 20-21, and 29-34 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____
- 4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date _____
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: _____

Detailed Action

Continued Examination Under 37 CFR 1.114

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(c), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(c) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 4/30/09 has been entered.

Status of claims

2. Claims 1-4, 6-11, 13-18, 20, 21 and 29-34 are pending.
3. Claims 1-4, 6-11, 13-18, 20, 21 and 29-34 have been examined.

Claim Rejections - 35 USC § 112

4. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

Claims 1, 8, and 15 are rejected under 35 U.S.C. 112, second paragraph, as being incomplete for omitting essential elements, such omission amounting to a gap between the elements. See MPEP § 2172.01. The omitted elements are: what do the handler's do with the code fragments.

Claim 8 recites the limitation "the providing permits" in claim 8 line 8 and 10. There is insufficient antecedent basis for this limitation in the claim.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

6. Claims 1-4, 6-11, 13-18, 20, 21 and 29-34 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent 6,874,143 by Murray et. al. (hereafter Murray) further in view of U.S. Patent 6,106,569 by Bohrer et. al. (hereafter Bohrer), and U.S. Patent 6,958,666 by Lowry et. al. (hereafter Lowry).

Claim 1:

Murray discloses the following claimed limitations:

"a user interface;"[col. 6 line 66, user interface. Accordingly, a user interface (user interface)is disclosed]

"a plurality of services, wherein a service includes a public interface that has an implementation and provides access to functionality in an extension."[col. 7 lines 12-13, An

EDF, implemented in XML, contains various tags that are associated with various extensions.

Col. 7 lines 21-25, EDFs advantageously have an “open schema” which means that third party developers can extend the extension mechanism and include their own extensions by creating their own tags. Col. 7 line 27-29, exemplary predefined XML tags for user interface elements can include tags for feature types such as: tool bars, accelerators, menu items, and themes.

Accordingly, a plurality of services (tags), wherein a service includes a public interface that has an implementation (open schema) and provides access to functionality in an extension (feature types such as tool bars, accelerators, menu items and themes) is disclosed.]

“a plurality of extensions to extend an application;”[abstract, various extensions can be developed by third party developers for incorporation into the program or platform. Accordingly, a plurality of extensions (various extensions) to extend an application (incorporation into the program or platform) is disclosed.]

wherein an extension includes “, and an XML (Extensible Markup Language) description” [col. 6 lines 24-25, extensions are described using XML]

“wherein each one of the plurality of services is associated with an extension in the plurality of extensions;”[col. 7 lines 12-13, an edf, implemented in XML, contains various tags that are associated with various extensions. Col. 7 lines 14-21, user interface elements, behaviors/component objects, store elements, user-defined objects, or anything else that represents a point of extensibility in the application or platform. Accordingly, wherein each one of the plurality of services (tags) is associated with an extension (user interface elements) in the

plurality of extensions (user interface elements, behaviors/component objects, store elements, user-defined objects, or anything else that represents a point of extensibility in the application or platform) is disclosed.]

“wherein one of the plurality of extensions exposes and consumes services associated with another extension in the plurality of extensions;”[col. 22 lines 45-46, receives all of the EDFs and merges them together and exposes them as a single list. Accordingly, wherein one of the plurality of extensions exposes (exposes) and consumes services (merges) associated with another extension in the plurality of extensions (EDFs).]

“wherein one of the plurality of extensions provides functionality accessible in the user interface; and”[col. 6 lines 64-66, an EDF is an XML file that logically describes an extension. For example, the EDF can describe HTML that makes up a user interface. Accordingly, wherein one of the plurality of extensions (extension) provides functionality accessible in the user interface (user interface).]

“wherein one of the plurality of services provides access to functionality in one of the plurality of extensions.” [col. 7 lines 11-21, an EDF, implemented in XML, contains various tags that are associated with various extensions. The various tags can correspond to: user interface elements, behaviour/component/objects, store elements, user-defined objects, or anything else that represents a point of extensibility in the application or platform. Accordingly, wherein one of the plurality of services (tags) provides access to functionality in one of the plurality of extensions (extensions).]

Murray does not explicitly disclose "wherein an extension includes a set of classes defined in an object-oriented programming language" and "wherein the services are consumed by the set of classes"

On the other hand, Bohrer discloses

"wherein an extension includes a set of classes defined in an object-oriented programming language" [col. 1 lines 66-67, object oriented technology provides a class as a kind of software template from which individual "objects" can be instantiated. Col. 8 lines 51-53, when the pickable extension 502 is added to the order item, it then creates an adaptor object 503 which inherits from the polymorphically used pickable base class 404. Accordingly, an extension (e.g. pickable extension) includes a set of classes (e.g. base class) defined in an object oriented programming language (adaptor object)]

"wherein the services are consumed by the set of classes" [when the pickable extension is added to the order item it then creates an adaptor object 503 which inherits from the polymorphically used "pickable" base class...if the "print pick list" method is subsequently blocked, the pickable extension would delete its adaptor. The ability Accordingly, wherein the services (print pick list method) are consumed by the set of classes (adaptor object inherits from the polymorphically used "pickable" base class)]

Both Murray and Bohrer disclose methods in which to extend applications. Both are within the same field of endeavor as applicant's invention, as they are both directed to providing extensions to an application. Murray discloses extensions. Bohrer further discloses an extensible item

feature allowing an interface of the object to be broken into logically dynamically changing pieces. It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to have applied Bohrer's disclosure above to the disclosure of Murray for the purpose of allowing behavior of the extensions to be replaced or overridden.

Murray and Bohrer do not explicitly disclose “, wherein the XML description is scanned for code fragments to be passed to handlers defined for a particular ID attribute”

On the other hand, Lowry discloses abstract, an XML file as an input stream, parses the input stream, and scans the input stream for an object. Upon finding an object, the method determines whether the object references a system service and dynamically loads the referenced service.

Accordingly, Lowry discloses wherein the XML description (XML file) is scanned (scans) for code fragments (objects) to be passed to handlers (system service) defined for a particular ID attribute (object references a system service).

Murray, Bohrer, and Lowry are all within the same field of endeavor as applicant's invention. Murray, Bohrer, and Lowry are all disclose methods of application development. The Bohrer provides an extensible item feature allowing an interface of the object to be broken into logically dynamically changing pieces. Murray discloses that an extension is described using XML. Combining Bohrer with Murray improves upon Murray by allowing for extensions to be replaced or overridden. However, Bohrer and Murray do not necessarily disclose scanning the

XML to provide the extension. It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to apply Lowry's disclosure above to the combination of Bohrer and Murray for the purpose of scanning the xml to provide the system services in order to provide the extension that an application requires.

Claim 2:

The combination of Murray, Bohrer, and Lowry disclose in Murray the following claimed limitations:

“One of the plurality of extensions utilizes one of the plurality of services” [col. 7 lines 11-21].

Claim 3:

The combination of Murray, Bohrer, and Lowry disclose in Murray the following claimed limitations:

“an extension is an interchangeable application building block” [col. 7 lines 25-26, extensions can themselves be extended by other developers].

Claim 4:

The combination of Murray, Bohrer, and Lowry disclose in Murray discloses the following claimed limitations:

“the extension includes a set of resources” [col. 7 lines 63-64, classes and offline data sources.].

Claim 6:

The combination of Murray, Bohrer, and Lowry disclose in Murray the following claimed limitations:

“an extension defines handlers for an XML description tag found in the XML description” [col. 7 lines 1-2, the EDF can also contain all or part of the functionality that comprises an extension. Col. 7 lines 45-46, exemplary predefined XML tags for behaviors/components/objects include tags for services. Accordingly, disclosing an extension (extension) defines handlers (services) for an xml tag (tag) found in the xml description (xml)]

Claim 7:

The combination of Murray, Bohrer, and Lowry disclose in Murray the following claimed limitations:

“an extension provides functionality to support at least one of: 1) a document type; 2) a user interface action; 3) a file encoding; 4) property settings; and 5) debugging information” [col. 6 lines 65-66, For example the EDF can describe HTML that makes up a user interface.].

Claim 8:

Murray discloses the following claimed limitations:

“providing a user interface to allow user interaction with the application,” [col. 6 line 66, user interface. Accordingly, providing a user interface to allow user interaction with the application (user interface)]

“providing a plurality of extensions to extend the application,” [abstract, various extensions can be developed by third party developers for incorporation into the program or platform. Accordingly, providing a plurality of extensions (various extensions) to extend an application (incorporation into the program or platform).]

wherein an extension includes “, and an XML (Extensible Markup Language) description” [col. 6 lines 24-25, extensions are described using XML]

“wherein the providing permits one of the plurality of extensions to provide functionality accessible in the user interface; and” [col. 6 lines 64-66, an EDF is an XML file that logically describes an extension. For example, the EDF can describe HTML that makes up a user interface. col. 7 lines 12-13, an edf, implemented in XML, contains various tags that are associated with various extensions. Col. 7 lines 14-21, user interface elements, behaviors/component objects, store elements, user-defined objects, or anything else that represents a point of extensibility in the application or platform. Accordingly, wherein the providing permits one of the plurality of extensions (extension) provides functionality accessible in the user interface (user interface)is disclosed.]

“providing a plurality of services wherein the providing permits one of the plurality of services to provide access to functionality in one of the plurality of extensions;” [col. 7 lines 11-21, an EDF, implemented in XML, contains various tags that are associated with various extensions. The various tags can correspond to: user interface elements, behaviour/component/objects, store elements, user-defined objects, or anything else that represents a point of extensibility in the application or platform. Accordingly, providing a plurality of services wherein the providing permits one of the plurality of services (tags) to provide access to functionality in one of the plurality of extensions (extensions).]

“wherein a service includes a public interface that has an implementation and provides access to functionality in an extension;” [col. 7 lines 12-13, An EDF, implemented in XML,

contains various tags that are associated with various extensions. Col. 7 lines 21-25, EDFs advantageously have an “open schema” which means that third party developers can extend the extension mechanism and include their own extensions by creating their own tags. Col. 7 line 27-29, exemplary predefined XML tags for user interface elements can include tags for feature types such as: tool bars, accelerators, menu items, and themes. Accordingly, wherein a service includes a public interface that has an implementation (open schema) and provides access to functionality in an extension (feature types such as tool bars, accelerators, menu items and themes) is disclosed.]

“wherein each one of the plurality of services is associated with an extension in the plurality of extensions; and” [col. 7 lines 12-13, an edf, implemented in XML, contains various tags that are associated with various extensions. Col. 7 lines 14-21, user interface elements, behaviors/component objects, store elements, user-defined objects, or anything else that represents a point of extensibility in the application or platform. Accordingly, wherein each one of the plurality of services (tags) is associated with an extension (user interface elements) in the plurality of extensions (user interface elements, behaviors/component objects, store elements, user-defined objects, or anything else that represents a point of extensibility in the application or platform) is disclosed.]]

“wherein one of the plurality of extensions exposes and consumes services associated with another extension in the plurality of extensions.” [col. 22 lines 45-46, receives all of the EDFs and merges them together and exposes them as a single list. Accordingly, wherein one of the plurality of extensions exposes (exposes) and consumes services (merges) associated with another extension in the plurality of extensions (EDFs).]

Murray does not explicitly disclose “wherein an extension includes a set of classes defined in an object-oriented programming language” and “wherein the services are consumed by the set of classes”

On the other hand, Bohrer discloses

“wherein an extension includes a set of classes defined in an object-oriented programming language” [col. 1 lines 66-67, object oriented technology provides a class as a kind of software template from which individual “objects” can be instantiated. Col. 8 lines 51-53, when the pickable extension 502 is added to the order item, it then creates an adaptor object 503 which inherits from the polymorphically used pickable base class 404. Accordingly, an extension (e.g. pickable extension) includes a set of classes (e.g. base class) defined in an object oriented programming language (adaptor object)]

“wherein the services are consumed by the set of classes” [when the pickable extension is added to the order item it then creates an adaptor object 503 which inherits from the polymorphically used “pickable” base class...if the “print pick list” method is subsequently blocked, the pickable extension would delete its adaptor. The ability Accordingly, wherein the services (print pick list method) are consumed by the set of classes (adaptor object inherits from the polymorphically used “pickable” base class)]

Both Murray and Bohrer disclose methods in which to extend applications. Both are within the same field of endeavor as applicant's invention, as they are both directed to providing extensions

to an application. It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to have applied Bohrer's disclosure above to the disclosure of Murray for the purpose of allowing behavior of the extensions to be replaced or overridden.

Murray and Bohrer do not explicitly disclose “, wherein the XML description is scanned for code fragments to be passed to handlers defined for a particular ID attribute”

On the other hand, Lowry discloses abstract, an XML file as an input stream, parses the input stream, and scans the input stream for an object. Upon finding an object, the method determines whether the object references a system service and dynamically loads the referenced service.

Accordingly, Lowry discloses wherein the XML description (XML file) is scanned (scans) for code fragments (objects) to be passed to handlers (system service) defined for a particular ID attribute (object references a system service).

Murray, Bohrer, and Lowry are all within the same field of endeavor as applicant's invention. Murray, Bohrer, and Lowry all disclose methods of application development. The Bohrer provides an extensible item feature allowing an interface of the object to be broken into logically dynamically changing pieces. Murray discloses that an extension is described using XML. Combining Bohrer with Murray improves upon Murray by allowing for extensions to be replaced or overridden. However, Bohrer and Murray do not necessarily disclose scanning the XML to provide the extension. It would have been obvious to a person of an ordinary skill in the

art at the time the invention was made to apply Lowry's disclosure above to the combination of Bohrer and Murray for the purpose of scanning the xml to provide the system services in order to provide the extension that an application requires.

Claim 9:

The combination of Murray, Bohrer, and Lowry disclose in Murray the following claimed limitations:

“one of the plurality of extensions utilizes one of the plurality of services.” [col. 7 lines 11-21].

Claim 10:

The combination of Murray, Bohrer, and Lowry disclose in Murray the following claimed limitations:

“an extension is an interchangeable application building block.” [col. 7 lines 25-26, extensions can themselves be extended by other developers].

Claim 11:

The combination of Murray, Bohrer, and Lowry disclose in Murray the following claimed limitations:

“an extension includes a set of resources” [col. 7 lines 63-64, classes and offline data sources.].

Claim 13:

The combination of Murray, Bohrer, and Lowry disclose in Murray the following claimed limitations:

“an extension defines handlers for an XML tag found in the XML description.” [col. 7 lines 1-2, the EDF can also contain all or part of the functionality that comprises an extension. Col. 7 lines 45-46, exemplary predefined XML tags for behaviors/components/objects include tags for services. Accordingly, discloses an extension (extension) defines handlers (services) for an xml tag (tag) found in the xml description (xml)]

Claim 14:

The combination of Murray, Bohrer, and Lowry disclose in Murray the following claimed limitations:

“an extension provides functionality to support at least one of: 1) a document type; 2) a user interface action; 3) a file encoding; 4) property settings; and 5) debugging information.” [col. 6 lines 65-66, For example the EDF can describe HTML that makes up a user interface.].

Claim 15:

Murray discloses the following claimed limitations:

“provide a user interface to allow user interaction with an application;” [col. 6 line 66, user interface. Accordingly, provide a user interface to allow user interaction with the application (user interface)is disclosed]

“provide a plurality of extensions to extend the application,” [abstract, various extensions can be developed by third party developers for incorporation into the program or platform. Accordingly, provide a plurality of extensions (various extensions) to extend an application (incorporation into the program or platform)is disclosed.] “wherein the providing permits one of

the plurality of extensions to provide functionality accessible in the user interface; and” [col. 6 lines 64-66, an EDF is an XML file that logically describes an extension. For example, the EDF can describe HTML that makes up a user interface. Accordingly, wherein the providing permits one of the plurality of extensions (extensions) provides functionality accessible in the user interface (user interface)is disclosed.]

wherein an extension includes “, and an XML (Extensible Markup Language) description” [col. 6 lines 24-25, extensions are described using XML]

“provide a plurality of services wherein the providing permits one of the plurality of services to provide access to functionality in one of the plurality of extensions;” [col. 7 lines 11-21, an EDF, implemented in XML, contains various tags that are associated with various extensions. The various tags can correspond to: user interface elements, behaviour/component/objects, store elements, user-defined objects, or anything else that represents a point of extensibility in the application or platform. Accordingly, provide a plurality of services (tags) wherein the providing permits one of the plurality of services (tags) to provide access to functionality in one of the plurality of extensions (extensions).]

“wherein a service includes a public interface that has an implementation and provides access to functionality in an extension;” [col. 7 lines 12-13, An EDF, implemented in XML, contains various tags that are associated with various extensions. Col. 7 lines 21-25, EDFs advantageously have an “open schema” which means that third party developers can extend the extension mechanism and include their own extensions by creating their own tags. Col. 7 line 27-29, exemplary predefined XML tags for user interface elements can include tags for feature types such as: tool bars, accelerators, menu items, and themes. Accordingly, wherein a service

includes a public interface that has an implementation (open schema) and provides access to functionality in an extension (feature types such as tool bars, accelerators, menu items and themes)is disclosed.]

“wherein each one of the plurality of services is associated with an extension in the plurality of extensions; and” [col. 7 lines 12-13, an edf, implemented in XML, contains various tags that are associated with various extensions. Col. 7 lines 14-21, user interface elements, behaviors/component objects, store elements, user-defined objects, or anything else that represents a point of extensibility in the application or platform. Accordingly, wherein each one of the plurality of services (tags) is associated with an extension (user interface elements) in the plurality of extensions (user interface elements, behaviors/component objects, store elements, user-defined objects, or anything else that represents a point of extensibility in the application or platform) is disclosed.]]

“wherein one of the plurality of extensions exposes and consumes services associated with another extension in the plurality of extensions.” [col. 7 lines 1-2, EDF can also contain all or part of the functionality that comprises an extension. col. 22 lines 45-46, receives all of the EDFs and merges them together and exposes them as a single list. Accordingly, wherein one of the plurality of extensions exposes (exposes) and consumes services (merges) associated with another extension in the plurality of extensions (extension).]

Murray does not explicitly disclose “wherein an extension includes a set of classes defined in an object-oriented programming language” and “wherein the services are consumed by the set of classes”

On the other hand, Bohrer discloses

“wherein an extension includes a set of classes defined in an object-oriented programming language” [col. 1 lines 66-67, object oriented technology provides a class as a kind of software template from which individual "objects" can be instantiated. Col. 8 lines 51-53, when the pickable extension 502 is added to the order item, it then creates an adapter object 503 which inherits from the polymorphically used pickable base class 404. Accordingly, an extension (e.g. pickable extension) includes a set of classes (e.g. base class) defined in an object oriented programming language (adaptor object)]

“wherein the services are consumed by the set of classes” [when the pickable extension is added to the order item it then creates an adaptor object 503 which inherits from the polymorphically used "pickable" base class...if the "print pick list" method is subsequently blocked, the pickable extension would delete its adaptor. The ability Accordingly, wherein the services (print pick list method) are consumed by the set of classes (adaptor object inherits from the polymorphically used “pickable” base class)]

Both Murray and Bohrer disclose methods in which to extend applications. Both are within the same field of endeavor as applicant's invention, as they are both directed to providing extensions to an application. It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to have applied Bohrer's disclosure above to the disclosure of Murray for the purpose of allowing behavior of the extensions to be replaced or overridden.

Murray and Bohrer do not explicitly disclose “, wherein the XML description is scanned for code fragments to be passed to handlers defined for a particular ID attribute”

On the other hand, Lowry discloses abstract, an XML file as an input stream, parses the input stream, and scans the input stream for an object. Upon finding an object, the method determines whether the object references a system service and dynamically loads the referenced service.

Accordingly, Lowry discloses wherein the XML description (XML file) is scanned (scans) for code fragments (objects) to be passed to handlers (system service) defined for a particular ID attribute (object references a system service).

Murray, Bohrer, and Lowry are all within the same field of endeavor as applicant's invention. Murray, Bohrer, and Lowry are all disclose methods of application development. The Bohrer provides an extensible item feature allowing an interface of the object to be broken into logically dynamically changing pieces. Murray discloses that an extension is described using XML. Combining Bohrer with Murray improves upon Murray by allowing for extensions to be replaced or overridden. However, Bohrer and Murray do not necessarily disclose scanning the XML to provide the extension. It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to apply Lowry's disclosure above to the combination of Bohrer and Murray for the purpose of scanning the xml to provide the system services in order to provide the extension that an application requires.

Claim 16:

The combination of Murray, Bohrer, and Lowry disclose in Murray the following claimed limitations:

“one of the plurality of extensions utilizes one of the plurality of services.” [col. 7 lines 11-21]

Claim 17:

The combination of Murray, Bohrer, and Lowry disclose in Murray the following claimed limitations:

“an extension is an interchangeable application building block.” [col. 7 lines 25-26, extensions can themselves be extended by other developers]

Claim 18:

The combination of Murray, Bohrer, and Lowry disclose in Murray the following claimed limitations:

“an extension includes an XML (extensible markup language) description” [col. 6 lines 64-66, EDFs] “and at least one of: 1) a set of classes; and 2) a set of resources” [col. 7 lines 63-64, classes and offline data sources.].

Claim 20:

The combination of Murray, Bohrer, and Lowry disclose in Murray the following claimed limitations:

“an extension defines handlers for an xml tag found in the xml description.” [col. 7 lines 1-2, the EDF can also contain all or part of the functionality that comprises an extension. Col. 7 lines 45-46, exemplary predefined XML tags for behaviors/components/objects include tags for

services. Accordingly, discloses an extension (extension) defines handlers (services) for an xml tag (tag) found in the xml description (xml)]

Claim 21:

The combination of Murray, Bohrer, Lowry disclose in Murray the following claimed limitations:

“an extension provides functionality to support at least one of: 1) a document type; 2) a user interface action; 3) a file encoding; 4) property settings; and 5) debugging information.”
[col. 6 lines 65-66, For example the EDF can describe HTML that makes up a user interface.]

Claim 29 :

The combination of Murray, Bohrer, and Lowry disclose in Murray “, wherein the plurality of services includes at least one of:” (col. 7 lines 55-60, services)

“a resource service to provide access to a set of resources;”

“a frame service to allow extensions to specify a graphical user interface (GUI) docking layout;”

“a file service to provide a set of services for file system access and manipulation;”

“a server service to provide a set of services for accessing a server;”

“a document service to supply an abstract document interface for files that are part of an application project; and”(col. 7 lines 55-60, for example editing functions use object model attachments attached to the window or document that maintain document context and editing state per window. These can also include object model attachments attached to the application. Accordingly, document service (editing functions use object model attachments) to supply an abstract document interface for files (object model attachments) that are a part of an application

(object model attachments attached to application))

“an action service to provide methods for adding and manipulating menu and toolbar items.”

Claim 30 :

The combination of Murray, Bohrer, and Lowry disclose in Murray “, wherein the plurality of services includes at least one of:” (col. 7 lines 55-60, services)

“a resource service to provide access to a set of resources;”

“a frame service to allow extensions to specify a graphical user interface (GUI) docking layout;”

“a file service to provide a set of services for file system access and manipulation;”

“a server service to provide a set of services for accessing a server;”

“a document service to supply an abstract document interface for files that are part of an application project; and”(col. 7 lines 55-60, for example editing functions use object model attachments attached to the window or document that maintain document context and editing state per window. These can also include object model attachments attached to the application. Accordingly, document service (editing functions use object model attachments) to supply an abstract document interface for files (object model attachments) that are a part of an application (object model attachments attached to application))

“an action service to provide methods for adding and manipulating menu and toolbar items.”

Claim 31 :

The combination of Murray, Bohrer, and Lowry disclose in Murray “, wherein the plurality of services includes at least one of:” (col. 7 lines 55-60, services)

“a resource service to provide access to a set of resources;”

“a frame service to allow extensions to specify a graphical user interface (GUI) docking layout;”

“a file service to provide a set of services for file system access and manipulation;”

“a server service to provide a set of services for accessing a server;”

“a document service to supply an abstract document interface for files that are part of an application project; and”(col. 7 lines 55-60, for example editing functions use object model attachments attached to the window or document that maintain document context and editing state per window. These can also include object model attachments attached to the application. Accordingly, document service (editing functions use object model attachments) to supply an abstract document interface for files (object model attachments) that are a part of an application (object model attachments attached to application))

“an action service to provide methods for adding and manipulating menu and toolbar items.”

Claim 32 :

The combination of Murray, Bohrer, and Lowry disclose in Lowry, “wherein the code fragments are contained within an XML tag and the code fragments are passed to handlers for the particular ID attribute at runtime.”[See figure 5; see abstract, the method accepts an XML file as an input stream, parses the input stream and scans the input stream for an object. Upon finding an object, the method determines whether the object references a system service and dynamically loads the referenced service. Accordingly, wherein the code fragments (object) are contained within an XML tag (XML file may contain a set of nested statements) and the code fragments (objects) are

passed to handlers (system services) for the particular ID attribute (object references) at runtime (figure 6).]

Claim 33 :

The combination of Murray, Bohrer, and Lowry disclose in Lowry, “wherein the code fragments are contained within an XML tag and the code fragments are passed to handlers for the particular ID attribute at runtime.”[See figure 5; see abstract, the method accepts an XML file as an input stream, parses the input stream and scans the input stream for an object. Upon finding an object, the method determines whether the object references a system service and dynamically loads the referenced service. Accordingly, wherein the code fragments (object) are contained within an XML tag (XML file may contain a set of nested statements) and the code fragments (objects) are passed to handlers (system services) for the particular ID attribute (object references) at runtime (figure 6).]

Claim 34 :

The combination of Murray, Bohrer, and Lowry disclose in Lowry, “wherein the code fragments are contained within an XML tag and the code fragments are passed to handlers for the particular ID attribute at runtime.”[See figure 5; see abstract, the method accepts an XML file as an input stream, parses the input stream and scans the input stream for an object. Upon finding an object, the method determines whether the object references a system service and dynamically loads the referenced service. Accordingly, wherein the code fragments (object) are contained within an XML tag (XML file may contain a set of nested statements) and the code fragments (objects) are

passed to handlers (system services) for the particular ID attribute (object references) at runtime (figure 6).]

Response to Arguments

7. Applicant's arguments with respect to claims 1-4, 6-11, 13-18, 20, 21 and 29-34 have been considered but are moot in view of the new ground(s) of rejection.

With respect to the Murray reference applicant's assert the following:

A. That Murray and Boherer do not disclose "an extension that includes 'an XML (Extensible Markup Language) description, wherein the XML description is scanned for code fragments to be passed to handlers defined for a particular ID attribute"

In response, this is moot.

B. That none of the cited prior art is directed to a computer-based extendable application framework.

In response, none of the claimed limitations recited in the body of the claim are in regard to an application framework.

Secondly, all references are directed to application frameworks. Murray, col. 21 lines 20-22, In the desired embodiment, one of the aspects that provide desirable utility is the extensibility

of the software platform. That is, third and fourth party developers are free to develop their own extensions which can be used within the framework of software platform. Bohrer, abstract, a method of developing a software system using object oriented technology and frameworks. Lowry, col. 1 lines 14-19, this invention relates generally to computer software and, more specifically to a system and method for providing distributed, directory-enabled applications using an extensible markup language (XML) application programming interface framework. Therefore, the statement that none of the cited prior art is directed to computer-based extendable application framework is respectfully disagreed.

Conclusion

8. The prior art made of record listed on PTO-892 and not relied, if any, upon is considered pertinent to applicant's disclosure.

Contact Information

9. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael D. Pham whose telephone number is (571)272-3924. The examiner can normally be reached on Monday - Friday 9am - 5:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Cottingham can be reached on 571-272-7079. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/M. P./
Examiner, Art Unit 2167

/John R. Cottingham/
Supervisory Patent Examiner, Art Unit
2167

